Fractal Decoded Image Quality Prediction Based on Accumulated Collage Error Coefficient

Qiang Wang

College of Information Science and Technology

Dalian Maritime University

Dalian, 116026, China

Email: wangqiang2011@dlmu.edu.cn

Abstract: To predict the fractal decoded image quality more efficiently, an accumulated collage error coefficient (ACEC) based method was proposed in this study. Firstly, the definition of ACEC was introduced to describe the relationship among the upper bound, lower bound, and actual value of the accumulated collage error of all range blocks. Moreover, the definition and monotonicity of the relative error of ACEC were also defined and discussed. While the relative error of ACEC reaches a relatively small value, the average collage error (ACER) can be estimated approximately, and then the encoding process can be terminated to directly predict the peak signal-to-noise ratio (PSNR) quality of decoded images. Experimental results show that compared with the previous method, the proposed method can predict the decoded image quality more accurately with fewer computations.

1. Introduction

Images play the most important role in human perception, and everyday a huge number of images are stored, processed, and transmitted, which will lead to great pressure to the transmission and storage requirements. Efficient image compression technology is an effective way to address the problem of reducing the amount of data. Different from conventional image compression techniques such as discrete cosine transform and discrete

wavelet transform, fractal image coding realizes image compression by establishing an iterated function system (IFS) whose fixed point can well approximate the original image. The kernel idea of fractal image coding was firstly introduced by Barnsley, and then Jacquin proposed the first block based fractal coding method[1,2]. Although fractal image coding has the advantages of novel idea, fast decoding, potential high compression ratio, and resolution independence[3,4], the drawback of fractal image coding was the high computational complexity in encoding process. Thus, fast fractal image encoding becomes one of the key research directions. In recent years, many fast fractal encoding methods were proposed by converting global block matching into local block matching[5-7]. For example, Chaurasia proposed composite statistical features, and Gupta extracted the features of the blocks in the DCT domain. The above two methods located the nearest neighbors in the feature space as the candidates of best-matched domain blocks, and then the block-matching operations were only carried out within the candidates. Instead of finding out the best matched domain block, Zheng specified the update times of fractal codes to obtain an acceptable domain block, and the encoding process can be effectively shortened. Further, the no-search fractal image coding methods directly assigned one domain block as the best-matched domain block for each range block without searching operations, and thus real-time encoding and higher compression ratio can be realized at the expense of poor decoded image quality[8-11]. After many years' development, fractal image coding has been gradually applied to other image processing applications, such as image denoising[12-18] , image retrieval[19-21], image super resolution[22-26], watermarking[27-29], image hashing[30,31], and head pose estimation[32,33].

The fractal decoded image quality prediction method was a recently proposed technique

which can predict the decoded image quality only with partial encoding process[34]. In the previous method, the average percentage of accumulated collage error is used to predict the decoded image quality, but the proportional relationship among the upper bound, lower bound, and actual value of accumulated collage error has not been considered. Thus, to predict the decoded image quality more efficiently, an accumulated collage error coefficient (ACEC) based method was proposed in this study. Based on the upper and lower bounds of the accumulated collage error of all range blocks, $\text{ACE}_{\text{Max}}^{\text{All}}$ and $\text{ACE}_{\text{Min}}^{\text{All}}$, the definition of ACEC was proposed to describe the proportional relationship among $\text{ACE}_{\text{Max}}^{\text{All}}$, $\text{ACE}_{\text{Min}}^{\text{All}}$, and the actual value of the accumulated collage error of all range blocks. Further, the definition and monotonicity of the relative error of ACEC were also defined and discussed. In encoding process, while the relative error of ACEC reaches a relatively small value, the encoding process can be terminated, and the average collage error (ACER) can be approximately estimated. Finally, based on the logarithmic relationship between ACER and the peak signal-to-noise ratio (PSNR) quality of decoded images, the decoded image quality can be directly predicted. Experimental results show that compared with the previous method, the proposed method can predict the PSNR quality of decoded images more accurately with fewer computations. The main contributions of this study can be summarized as:

1. Based on the upper and lower bounds of the accumulated collage error, the definition of accumulated collage error coefficient (ACEC) was proposed to describe the relationship among the upper bound, lower bound, and actual value of the accumulated collage error of all range blocks.

2. How to estimate ACEC, the definition and monotonicity of the relative error of ACEC

were also introduced and discussed, based on which the proposed ACEC based prediction method can provide higher prediction accuracy with fewer computations than the previous method.

3. The proposed method can be combined with the existing fractal encoding methods to predict the decoded image quality only with partial encoding process.

This paper is organized as follows: Conventional fractal image coding is reviewed in Section 2. The principle and detailed procedures of the proposed method are described in Section 3. The experimental results are presented and analyzed in Sections 4 and 5, respectively. The conclusion is given in Section 6.

2. Conventional fractal image coding

Fractal image coding aims to establish an IFS whose fixed point can approximately approach the original image. The first block-based fractal encoding method was proposed by Jacquin, and the detailed procedures can be described as[1]:

Step 1: Divide the $M \times N$ input image f uniformly into nonoverlapping $B \times B$ range blocks R_i, i=1,2,3,...,NumR, where NumR denotes the total number of range blocks.

Step 2: Slide a $2B \times 2B$ window over the original image horizontally and vertically, and the domain blocks D_j, j=1,2,3,...,NumD, can be obtained, where NumD denotes the total number of domain blocks.

Step 3: Establish a domain block pool (DBP) by contracting the domain blocks to the same size of range blocks. Then, an extended domain block pool (EDBP) can be obtained by performing eight isometric transformations on each domain block within DBP.

Step 4: The best matched domain block for each range block can be obtained by searching

4

within EDBP and minimizing the following function:

$$CE(\boldsymbol{R}_i) = \min_{\boldsymbol{D}_j}\left(\min_{s_i,o_i} \frac{1}{B^2}\left\| \boldsymbol{R}_i - s_i\boldsymbol{D}_j - o_i\boldsymbol{I} \right\|^2 \right)$$
$$i = 1,2,3,..., \text{NumR}, \ j = 1,2,3,..., \text{NumD}\times 8 \tag{1}$$

where CE($\boldsymbol{R}_i$) denotes the collage error of $\boldsymbol{R}_i$. $\boldsymbol{I}$ denotes a $B \times B$ matrix whose components are all ones. s_i and o_i denote the scaling and offset coefficients of the affine transformation, respectively. $\boldsymbol{D}_j$ is the best matched domain block of $\boldsymbol{R}_i$. Further, s_i and o_i can be computed with the least square method as:

$$s_i = \left\langle \boldsymbol{R}_i - \overline{r}_i\boldsymbol{I}, \ \ \boldsymbol{D}_j - \overline{d}_j\boldsymbol{I} \right\rangle \Big/ \left\| \boldsymbol{D}_j - \overline{d}_j\boldsymbol{I} \right\|^2, \ \ o_i = \overline{r}_i - s_i\overline{d}_j$$
$$i = 1,2,3,..., \text{NumR}, \ j = 1,2,3,..., \text{NumD}\times 8 \tag{2}$$

where $\langle \bullet,\bullet \rangle$ denotes the inner product. $\overline{r}_i$ and $\overline{d}_j$ denote the averages of $\boldsymbol{R}_i$ and $\boldsymbol{D}_j$, respectively.

In fractal decoding process, arbitrary $M \times N$ image can be selected as the initial image. Then, the same transformations as those established in encoding process are performed to reconstruct all range blocks until one iteration is completed. After about 10 iterations, the decoded image can be obtained. Figure 1 (a) illustrates the rotated 256×256 Bird image as the initial image, and Figure 2(b)-(f) illustrate the first five iteration images in decoding process while the Bird image is encoded in encoding process.
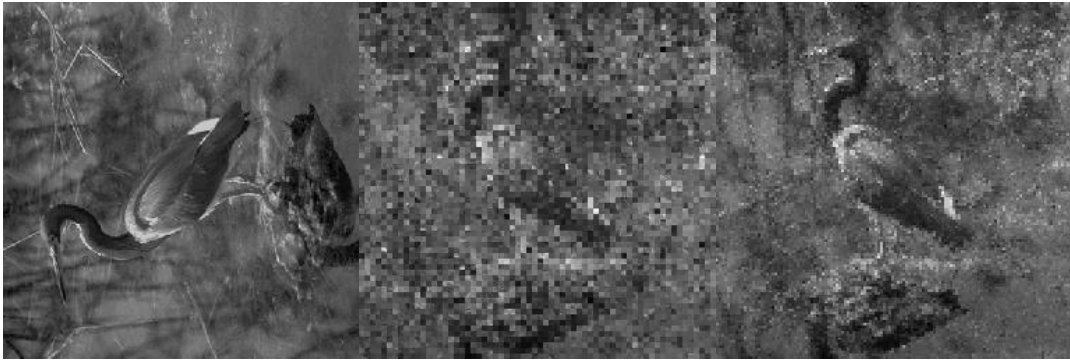
(a) Initial image   (b) First iteration   (c) Second iteration

| (d) Third iteration | (e) Fourth iteration | (f) Fifth iteration |

Fig.1 Initial and the first five iteration images in decoding process.

3. Prediction of the decoded image quality with the accumulated collage error coefficient.

3.1 Definition of the accumulated collage error coefficient

Substituting Eq.(2) back to (1) yields:

$$
\begin{aligned}
\mathrm{CE}(\boldsymbol{R}_i) &= \frac{1}{B^2}\left\| \boldsymbol{R}_i - s_i \boldsymbol{D}_j - o_i \boldsymbol{I} \right\|^2 = \frac{1}{B^2}\left(\left\| \boldsymbol{R}_i - \bar{r}_i \boldsymbol{I} \right\|^2 - s_i^2 \left\| \boldsymbol{D}_j - \bar{d}_j \boldsymbol{I} \right\|^2 \right) \\
&= \frac{1}{B^2}\left\| \boldsymbol{R}_i - \bar{r}_i \boldsymbol{I} \right\|^2 \left(1 - \frac{\left| \left\langle \boldsymbol{R}_i - \bar{r}_i \boldsymbol{I}, \boldsymbol{D}_j - \bar{d}_j \boldsymbol{I} \right\rangle \right|^2}{\left\| \boldsymbol{R}_i - \bar{r}_i \boldsymbol{I} \right\|^2 \left\| \boldsymbol{D}_j - \bar{d}_j \boldsymbol{I} \right\|^2} \right) = \frac{1}{B^2}\left\| \boldsymbol{R}_i - \bar{r}_i \boldsymbol{I} \right\|^2 \left(1 - \mathrm{LCC}_i^2 \right)
\end{aligned}
\tag{3}
$$

where LCC_i denote the linear correlation coefficient between $\boldsymbol{R}_i$ and $\boldsymbol{D}_j$. Because $0 \leq \mathrm{LCC}_i^2 \leq 1$, we have

$$
\mathrm{CE}\left(\boldsymbol{R}_i \right) \leq \frac{1}{B^2}\left\| \boldsymbol{R}_i - \bar{r}_i \boldsymbol{I} \right\|^2
\tag{4}
$$

From Eq.(4), we know that for arbitrary range block, the variance provides the upper bound for the associated collage error, and only the range blocks with large variances may have large collage errors. Thus, all range blocks are sorted by their variances from largest to smallest and encoded block by block. In this process, the accumulated collage error (ACE) of all range blocks can be described as:

6

$$\text{ACE}^{\text{All}} = \sum_{i=1}^{\text{NumR}} \text{CE}(\boldsymbol{R}_i) = \frac{1}{B^2} \sum_{i=1}^{\text{NumR}} \left\| \boldsymbol{R}_i - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\text{LCC}_i \right)^2 \right] \qquad (5)$$

Then, if we use LCC_{Min} and LCC_{Max} to represent the minimum and maximum of all

LCCs, the upper and lower bounds of ACE^{All} can be described as:

$$\begin{cases} \text{ACE}_{\text{Max}}^{\text{All}} = \sum_{i=1}^{\text{NumR}} \text{CE}(\boldsymbol{R}_i)_{\text{Max}} = \frac{1}{B^2} \sum_{i=1}^{\text{NumR}} \left\| \boldsymbol{R}_i - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\text{LCC}_{\text{Min}} \right)^2 \right] \\ \text{ACE}_{\text{Min}}^{\text{All}} = \sum_{i=1}^{\text{NumR}} \text{CE}(\boldsymbol{R}_i)_{\text{Min}} = \frac{1}{B^2} \sum_{i=1}^{\text{NumR}} \left\| \boldsymbol{R}_i - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\text{LCC}_{\text{Max}} \right)^2 \right] \end{cases} \qquad (6)$$

In encoding process, the range blocks can be divided into two categories: Coded range

blocks and uncoded range blocks. The ACE of coded range blocks can be represented as:

$$\text{ACE}^{\text{Coded}} = \sum_{i=1}^{\text{Num}^{\text{Coded}}} \text{CE}(\boldsymbol{R}_i^{\text{Coded}}) = \frac{1}{B^2} \sum_{i=1}^{\text{Num}^{\text{Coded}}} \left\| \boldsymbol{R}_i^{\text{Coded}} - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\text{LCC}_i^{\text{Coded}} \right)^2 \right] \qquad (7)$$

where $\text{Num}^{\text{Coded}}$ denote the number of coded range blocks. Similarly, the ACE of uncoded

range blocks can be represented as:

$$\text{ACE}^{\text{Uncoded}} = \sum_{i=1}^{\text{Num}^{\text{Uncoded}}} \text{CE}(\boldsymbol{R}_i^{\text{Uncoded}}) = \frac{1}{B^2} \sum_{i=1}^{\text{Num}^{\text{Uncoded}}} \left\| \boldsymbol{R}_i^{\text{Uncoded}} - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\text{LCC}_i^{\text{Uncoded}} \right)^2 \right] \qquad (8)$$

where $\text{Num}^{\text{Uncoded}}$ denote the number of uncoded range blocks. If we use $\text{LCC}_{\text{Max}}^{\text{Uncoded}}$ and

$\text{LCC}_{\text{Min}}^{\text{Uncoded}}$ to represent the maximum and minimum of the LCCs of uncoded range blocks,

the upper and lower bounds of the ACE of uncoded range blocks, $\text{ACE}_{\text{Max}}^{\text{Uncoded}}$ and

$\text{ACE}_{\text{Min}}^{\text{Uncoded}}$, can be expressed as:

$$\begin{cases} \text{ACE}_{\text{Max}}^{\text{Uncoded}} = \sum_{i=1}^{\text{Num}^{\text{Uncoded}}} \text{CE}(\boldsymbol{R}_i^{\text{Uncoded}})_{\text{Max}} = \frac{1}{B^2} \sum_{i=1}^{\text{Num}^{\text{Uncoded}}} \left\| \boldsymbol{R}_i^{\text{Uncoded}} - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\text{LCC}_{\text{Min}}^{\text{Uncoded}} \right)^2 \right] \\ \text{ACE}_{\text{Min}}^{\text{Uncoded}} = \sum_{i=1}^{\text{Num}^{\text{Uncoded}}} \text{CE}(\boldsymbol{R}_i^{\text{Uncoded}})_{\text{Min}} = \frac{1}{B^2} \sum_{i=1}^{\text{Num}^{\text{Uncoded}}} \left\| \boldsymbol{R}_i^{\text{Uncoded}} - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\text{LCC}_{\text{Max}}^{\text{Uncoded}} \right)^2 \right] \end{cases} \qquad (9)$$

Then, we have the inequality as:

$$\text{ACE}_{\text{Min}}^{\text{Uncoded}} < \text{ACE}^{\text{Uncoded}} < \text{ACE}_{\text{Max}}^{\text{Uncoded}} \qquad (10)$$

Moreover, because the uncoded range blocks are the subset of all range blocks, $\text{LCC}_{\text{Max}}^{\text{Uncoded}}$

and $LCC_{Min}^{Uncoded}$ can be approximated by LCC_{Max} and LCC_{Min}, respectively. Then, Eq.(9) can be rewritten by replacing $LCC_{Max}^{Uncoded}$ and $LCC_{Min}^{Uncoded}$ with LCC_{Max} and LCC_{Min}, respectively, as:

$$\begin{cases} ACE_{Max}^{Uncoded} = \sum_{i=1}^{Num^{Uncoded}} CE(\boldsymbol{R}_i^{Uncoded})_{Max} = \dfrac{1}{B^2} \sum_{i=1}^{Num^{Uncoded}} \left\| \boldsymbol{R}_i^{Uncoded} - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(LCC_{Min} \right)^2 \right] \\ ACE_{Min}^{Uncoded} = \sum_{i=1}^{Num^{Uncoded}} CE(\boldsymbol{R}_i^{Uncoded})_{Min} = \dfrac{1}{B^2} \sum_{i=1}^{Num^{Uncoded}} \left\| \boldsymbol{R}_i^{Uncoded} - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(LCC_{Max} \right)^2 \right] \end{cases} \quad (11)$$

For all range blocks, to describe the proportional relationship among ACE^{All}, ACE_{Max}^{All} and ACE_{Min}^{All}, the ratio of ACE_{Max}^{All} to ACE^{All} ratio to ACE^{All} to ACE_{Min}^{All} ratio is defined as the accumulated collage error coefficient, $ACEC^{All}$, which can be calculated by:

$$ACEC^{All} = \frac{ACE_{Max}^{All} / ACE^{All}}{ACE^{All} / ACE_{Min}^{All}} = \frac{ACE_{Max}^{All} ACE_{Min}^{All}}{\left(ACE^{All} \right)^2} \quad (12)$$

Further, Eq.(12) can be expanded as:

$$ACEC^{All} = \frac{ACE_{Max}^{All} ACE_{Min}^{All}}{\left(ACE^{Coded} + ACE^{Uncoded} \right)^2} \quad (13)$$

Further, by Eq.(10), the upper and lower bounds of $ACEC^{All}$, $ACEC_{Max}^{All}$ and $ACEC_{Min}^{All}$, can be represented, respectively, as:

$$\begin{cases} ACEC_{Max}^{All} = \dfrac{ACE_{Max}^{All} ACE_{Min}^{All}}{\left(ACE^{Coded} + ACE_{Min}^{Uncoded} \right)^2} \\ ACEC_{Min}^{All} = \dfrac{ACE_{Max}^{All} ACE_{Min}^{All}}{\left(ACE^{Coded} + ACE_{Max}^{Uncoded} \right)^2} \end{cases} \quad (14)$$

While $ACEC_{Max}^{All}$ can approach $ACEC_{Min}^{All}$ sufficiently, we can estimate $ACEC^{All}$ as:

$$\overline{ACEC}^{All} \approx \frac{ACEC_{Max}^{All} + ACEC_{Min}^{All}}{2} \quad (15)$$

3.2 Relative error of ACEC and Predicting the PSNR quality of decoded images

Based on the dynamic range of $ACEC^{All}$, i.e., the deviation between $ACEC_{Max}^{All}$ and

8

$ACEC_{Min}^{All}$, the relative error of $ACEC^{All}$, RE_{ACEC}, can be defined as the ratio of the half

deviation between $ACEC_{Max}^{All}$ and $ACEC_{Min}^{All}$ to the estimated $\overline{ACEC}^{All}$ by Eq.(15) as:

$$RE_{ACEC} = \frac{\left(ACEC_{Max}^{All} - ACEC_{Min}^{All}\right)\big/2}{\overline{ACEC}^{All}} \tag{16}$$

By Eqs.(15), (16) and (14), RE_{ACEC} can be expanded as:

$$
\begin{aligned}
RE_{ACEC} &= \frac{\left(ACEC_{Max}^{All} - ACEC_{Min}^{All}\right)\big/2}{\overline{ACEC}^{All}} = \frac{ACEC_{Max}^{All} - ACEC_{Min}^{All}}{ACEC_{Max}^{All} + ACEC_{Min}^{All}} \\
&= \frac{\left(ACE^{Coded} + ACE_{Max}^{Uncoded}\right)^2 - \left(ACE^{Coded} + ACE_{Min}^{Uncoded}\right)^2}{\left(ACE^{Coded} + ACE_{Max}^{Uncoded}\right)^2 + \left(ACE^{Coded} + ACE_{Min}^{Uncoded}\right)^2}
\end{aligned} \tag{17}
$$

Because a sufficiently large number of domain blocks within EDBP can guarantee that

$LCC_{Max} \approx 1$, $ACE_{Min}^{Uncoded} \approx 0$, and Eq.(17) can be rewritten as:

$$RE_{ACEC} = \frac{\left(ACE^{Coded} + ACE_{Max}^{Uncoded}\right)^2 - \left(ACE^{Coded}\right)^2}{\left(ACE^{Coded} + ACE_{Max}^{Uncoded}\right)^2 + \left(ACE^{Coded}\right)^2} \tag{18}$$

By setting the derivatives of RE_{ACEC} regarding ACE^{Coded} and $ACE_{Max}^{Uncoded}$ to zeros,

respectively, we have

$$
\begin{cases}
\dfrac{\partial RE_{ACEC}}{\partial ACE^{Coded}} = -4\left(ACE^{Coded}\right)\left(ACE_{Max}^{Uncoded}\right)\left(ACE^{Coded} + ACE_{Max}^{Uncoded}\right) < 0 \\
\dfrac{\partial RE_{ACEC}}{\partial ACE_{Max}^{Uncoded}} = 4\left(ACE^{Coded}\right)^2\left(ACE^{Coded} + ACE_{Max}^{Uncoded}\right) > 0
\end{cases} \tag{19}
$$

At the early stage of encoding process, ACE^{Coded} is much smaller than $ACE_{Max}^{Uncoded}$. By

Eq.(18), we know that $RE_{ACEC} \approx 1$. As the encoding process proceeds, because

$\dfrac{\partial RE_{ACEC}}{\partial ACE^{Coded}} < 0$ and $\dfrac{\partial RE_{ACEC}}{\partial ACE_{Max}^{Uncoded}} > 0$, RE_{ACEC} decreases monotonically while

ACE^{Coded} increases, and ACE^{Coded} decreases, respectively. At the final stage of the encoding

process, ACE^{Coded} is much larger than $ACE_{Max}^{Uncoded}$. By Eq.(18), we know that $RE_{ACEC} \approx 0$.

Thus, RE_{ACEC} set at a relatively small value can guarantee that a sufficiently large number

of range blocks have been encoded, and thus the maximum and minimum of the LCCs of coded range blocks, $\mathrm{LCC}_{\mathrm{Max}}^{\mathrm{Coded}}$ and $\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}}$, can be used to approximate those of all range blocks, $\mathrm{LCC}_{\mathrm{Max}}$ and $\mathrm{LCC}_{\mathrm{Min}}$, respectively, as:

$$\begin{cases} \mathrm{LCC}_{\mathrm{Max}}^{\mathrm{Coded}} \approx \mathrm{LCC}_{\mathrm{Max}} \\ \mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}} \approx \mathrm{LCC}_{\mathrm{Min}} \end{cases} \tag{20}$$

By Eq.(20), Eq.(6) can be rewritten by replacing $\mathrm{LCC}_{\mathrm{Max}}$ and $\mathrm{LCC}_{\mathrm{Min}}$ with $\mathrm{LCC}_{\mathrm{Max}}^{\mathrm{Coded}}$ and $\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}}$, respectively, as:

$$\begin{cases} \mathrm{ACE}_{\mathrm{Max}}^{\mathrm{All}} = \sum_{i=1}^{\mathrm{NumR}} \mathrm{CE}(\boldsymbol{R}_i)_{\mathrm{Max}} = \dfrac{1}{B^2} \sum_{i=1}^{\mathrm{NumR}} \left\| \boldsymbol{R}_i - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}} \right)^2 \right] \\ \mathrm{ACE}_{\mathrm{Min}}^{\mathrm{All}} = \sum_{i=1}^{\mathrm{NumR}} \mathrm{CE}(\boldsymbol{R}_i)_{\mathrm{Min}} = \dfrac{1}{B^2} \sum_{i=1}^{\mathrm{NumR}} \left\| \boldsymbol{R}_i - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\mathrm{LCC}_{\mathrm{Max}}^{\mathrm{Coded}} \right)^2 \right] \end{cases} \tag{21}$$

and similarly, Eq.(11) can be rewritten as:

$$\begin{cases} \mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Uncoded}} = \sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}})_{\mathrm{Max}} = \dfrac{1}{B^2} \sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \left\| \boldsymbol{R}_i^{\mathrm{Uncoded}} - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}} \right)^2 \right] \\ \mathrm{ACE}_{\mathrm{Min}}^{\mathrm{Uncoded}} = \sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}})_{\mathrm{Min}} = \dfrac{1}{B^2} \sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \left\| \boldsymbol{R}_i^{\mathrm{Uncoded}} - \overline{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\mathrm{LCC}_{\mathrm{Max}}^{\mathrm{Coded}} \right)^2 \right] \end{cases} \tag{22}$$

Thus, $\mathrm{ACEC}_{\mathrm{Max}}^{\mathrm{All}}$ and $\mathrm{ACEC}_{\mathrm{Min}}^{\mathrm{All}}$ in Eq.(14) can be obtained. Moreover, by Eq.(16), small $\mathrm{RE}_{\mathrm{ACEC}}$ can also guarantee that $\mathrm{ACEC}_{\mathrm{Max}}^{\mathrm{All}}$ is relatively close to $\mathrm{ACEC}_{\mathrm{Min}}^{\mathrm{All}}$. Then, $\overline{\mathrm{ACEC}}^{\mathrm{All}}$ can be estimated by Eq.(15), and $\mathrm{ACE}^{\mathrm{All}}$ can be computed by Eqs.(12) and (21) as:

$$\overline{\mathrm{ACE}}^{\mathrm{All}} = \sqrt{\dfrac{\mathrm{ACE}_{\mathrm{Max}}^{\mathrm{All}} \mathrm{ACE}_{\mathrm{Min}}^{\mathrm{All}}}{\overline{\mathrm{ACEC}}^{\mathrm{All}}}} \tag{23}$$

Further, the average collage error (ACER) of all range blocks can be computed as:

$$\mathrm{ACER} = \dfrac{\overline{\mathrm{ACE}}^{\mathrm{All}}}{\mathrm{NumR}} \tag{24}$$

In addition, the peak signal-to-noise ratio (PSNR) was used to measure the decoded image quality as:

$$\text{PSNR} = \log_{10}\left\{ 255^2 \Big/ \frac{1}{MN}\sum_{j=1}^{N}\sum_{i=1}^{M}\left[\boldsymbol{f}(i,j) - \boldsymbol{f}^{\text{Decoded}}(i,j)\right]^2 \right\} \qquad (25)$$

where $\boldsymbol{f}$ and $\boldsymbol{f}^{\text{Decoded}}$ represent the input and decoded images, respectively. Moreover, there exists a logarithmic relationship between ACER and PSNR as:

$$\text{PSNR} = \beta_1 + \beta_2\log_{10}(\text{ACER}) \qquad (26)$$

where β_1 and β_2 are both constant values. Thus, by Eqs.(23) and (24), we can directly predict the PSNR quality of the decoded image. The detailed procedures of the proposed method are listed as follows:

Step 1: Divide the input image into range and domain blocks, and sort the range blocks by their variances from largest to smallest.

Step 2: Take out one uncoded range block and encode it. Based on $\text{LCC}_{\text{Max}}^{\text{Coded}}$ and $\text{LCC}_{\text{Min}}^{\text{Coded}}$, $\text{ACEC}_{\text{Max}}^{\text{All}}$ and $\text{ACEC}_{\text{Min}}^{\text{All}}$ can be calculated by Eqs.(14), (21) and (22). Further, RE_{ACEC} can be computed by Eq.(16). If RE_{ACEC} is smaller than 0.15, go to Step 3. If not, turn back to Step 2.

Step 3: Calculate ACER by Eqs.(24), (23), (21), (15), and (14), and then predict the decoded image quality by Eq.(26).

4. Experiments

<div align="center">(a)Bird   (b) Owl   (c) Trees</div>

(d) City   (e) Cat

Fig.2 Five test images.

Five 256 × 256 images (in Fig.2), Bird, Owl, Trees, City, and Cat, are adopted in our experiment. Three state of the art fractal coding methods and Jacquin's method are used to assess the performance of the proposed method. The scaling and offset coefficients, s and o, are quantized by 5 and 7 bits, respectively. The previous prediction method is compared with the proposed method by prediction accuracy and the number of the computations required. Table 1, 2, 3, and 4 list the experimental results of Jacquin's, Chaurasia's, Zheng's, and Gupta's methods, respectively. In Tab.1, while the size of range blocks is 4 × 4, for Jacquin's method, the PSNRs of decoded images are listed in the second row. Moreover, in the third row, 4096 range blocks are needed to be encoded for all test images, and the percentages of the computations required (PCRs) are all considered as 100%. For the previous method, the fourth to sixth rows list the predicted PSNRs, deviations between the predicted and actual PSNRs, and PCRs. Similarly, the seventh to ninth rows show the counterparts for the proposed method. From Tab.1, we see that if the size of range blocks is 4 × 4, the average deviation decreases from 0.03dB to 0.02dB, the average PCR decreases from 62.72% and 56.20%. If the size of range blocks is 8 × 8, the average deviations decreases from 0.15dB to 0.12dB, the average PCR decreases from 62.81% and 54.47%. Thus, we know that for

Jacquin's method, the proposed method can provide higher prediction accuracy with fewer computations than the previous method.

Further, Table 2, 3, and 4 list the experimental results of Chaurasia's, Zheng's, and Gupta's methods, respectively. In Table 2, while the sizes of range blocks are set at 4×4 and 8×8, the average PCRs decrease from 62.98% to 56.46% and from 62.91% to 54.65%, respectively, and the average deviations decrease from 0.09dB to 0.07dB and from 0.18dB to 0.09dB, respectively. In Table 3, when the sizes of range blocks are set at 4×4 and 8×8, the average PCRs decrease from 70.97% to 64.64% and from 73.49% to 64.17%, respectively, and the average deviations remain at 0.05dB and decrease from 0.18dB to 0.10dB, respectively. In Table 4, when the sizes of range blocks are set at 4×4 and 8×8, the average PCRs decrease from 63.54% to 56.97% and from 63.11% to 54.86%, respectively, and the average deviations decrease from 0.08dB to 0.06dB and from 0.17dB to 0.11dB, respectively. In summary, the proposed method can predict the PSNR quality of decoded images more accurately with fewer computations than the previous method.

Tab.1 Performance comparison between the previous and proposed methods for Jacquin's method[1].

	Test images		Bird	Owl	Trees	City	Cat	Average
	Jacquin's[1]	Decoded(dB)	31.68	27.25	33.89	31.86	29.45	×
		Computations(%)	100	100	100	100	100	×
4 × 4	Previous[34]	Predicted(dB)	31.66	27.22	33.80	31.87	29.43	×
		Deviation(dB)	**0.02**	**0.03**	**0.09**	**0.01**	**0.02**	**0.03**
		Computations(%)	**68.99**	**72.73**	**71.17**	**50.90**	**49.80**	**62.72**
	Proposed	Predicted(dB)	31.67	27.24	33.82	31.87	29.45	×
		Deviation(dB)	**0.01**	**0.01**	**0.07**	**0.01**	**0.00**	**0.02▼**
		Computations(%)	**60.77**	**66.24**	**64.31**	**44.38**	**45.29**	**56.20▼**
8 × 8	Jacquin's[1]	Decoded(dB)	26.74	22.59	28.78	27.38	25.31	×
		Computations(%)	100	100	100	100	100	×
	Previous[34]	Predicted(dB)	26.63	22.37	28.62	27.30	25.11	×
		Deviation(dB)	**0.11**	**0.22**	**0.16**	**0.08**	**0.20**	**0.15**
		Computations(%)	**67.29**	**74.22**	**66.21**	**56.25**	**50.10**	**62.81**

		Bird	Owl	Trees	City	Cat	Average
Proposed	Predicted(dB)	26.65	22.40	28.69	27.31	25.16	×
	Deviation(dB)	**0.09**	**0.19**	**0.09**	**0.07**	**0.15**	**0.12▼**
	Computations(%)	**55.47**	**66.80**	**57.13**	**49.32**	**43.65**	**54.47▼**

Tab.2 Performance comparison between the previous and proposed methods for Chaurasia's method[5].

	Test images	Bird	Owl	Trees	City	Cat	Average
	Chaurasia's [5] Decoded(dB)	30.97	26.32	33.03	31.06	28.87	×
	Computations(%)	100	100	100	100	100	×
4×4	Previous[34] Predicted(dB)	30.96	26.25	32.90	30.94	28.77	×
	Deviation(dB)	**0.01**	**0.07**	**0.13**	**0.12**	**0.10**	**0.09**
	Computations(%)	**68.97**	**73.05**	**71.34**	**51.61**	**49.95**	**62.98**
	Proposed Predicted(dB)	30.97	26.27	32.92	30.94	28.80	×
	Deviation(dB)	**0.00**	**0.05**	**0.11**	**0.12**	**0.07**	**0.07▼**
	Computations(%)	**60.96**	**66.67**	**63.94**	**45.12**	**45.61**	**56.46▼**
	Chaurasia's [5] Decoded(dB)	26.56	22.43	28.61	27.17	25.11	×
	Computations(%)	100	100	100	100	100	×
8×8	Previous[34] Predicted(dB)	26.46	22.17	28.42	27.04	24.89	×
	Deviation(dB)	**0.10**	**0.26**	**0.19**	**0.13**	**0.22**	**0.18**
	Computations(%)	**66.89**	**74.12**	**67.97**	**55.76**	**49.80**	**62.91**
	Proposed Predicted(dB)	26.51	22.23	28.48	27.06	25.02	×
	Deviation(dB)	**0.05**	**0.20**	**0.13**	**0.11**	**0.09**	**0.09▼**
	Computations(%)	**55.18**	**66.80**	**59.28**	**48.73**	**43.26**	**54.65▼**

Tab.3 Performance comparison between the previous and proposed methods for Zheng's method[6].

	Test images	Bird	Owl	Trees	City	Cat	Average
	Zheng's[6] Decoded(dB)	30.86	26.52	33.04	31.03	28.94	×
	Computations(%)	100	100	100	100	100	×
4×4	Previous[34] Predicted(dB)	30.92	26.48	33.11	31.08	28.89	×
	Deviation(dB)	**0.06**	**0.04**	**0.07**	**0.05**	**0.05**	**0.05**
	Computations(%)	**79.87**	**90.77**	**76.59**	**55.69**	**51.92**	**70.97**
	Proposed Predicted(dB)	30.92	26.49	33.13	31.08	28.91	×
	Deviation(dB)	**0.06**	**0.03**	**0.09**	**0.05**	**0.03**	**0.05=**
	Computations(%)	**72.54**	**87.10**	**67.64**	**48.05**	**47.86**	**64.64▼**
	Zheng's[6] Decoded(dB)	26.43	22.27	28.45	27.02	24.97	×
	Computations(%)	100	100	100	100	100	×
8×8	Previous[34] Predicted(dB)	26.20	22.07	28.31	26.89	24.79	×
	Deviation(dB)	**0.23**	**0.20**	**0.14**	**0.13**	**0.18**	**0.18**
	Computations(%)	**79.11**	**89.58**	**76.88**	**68.17**	**53.70**	**73.49**
	Proposed Predicted(dB)	26.27	22.16	28.38	26.92	24.92	×
	Deviation(dB)	**0.16**	**0.11**	**0.07**	**0.10**	**0.05**	**0.10▼**
	Computations(%)	**63.90**	**83.09**	**67.37**	**59.61**	**46.88**	**64.17▼**

Tab.4 Performance comparison between the previous and proposed methods for Gupta's method[7].

	Test images		Bird	Owl	Trees	City	Cat	Average
	Gupta's[7]	Decoded(dB)	31.00	26.48	33.21	31.28	28.93	×
		Computations(%)	100	100	100	100	100	×
4×4	Previous[34]	Predicted(dB)	30.99	26.35	33.06	31.24	28.84	×
		Deviation(dB)	**0.01**	**0.13**	**0.15**	**0.04**	**0.09**	**0.08**
		Computations(%)	**69.48**	**74.00**	**72.39**	**51.71**	**50.12**	**63.54**
	Proposed	Predicted(dB)	31.01	26.38	33.11	31.25	28.88	×
		Deviation(dB)	**0.01**	**0.10**	**0.10**	**0.03**	**0.05**	**0.06▼**
		Computations(%)	**61.60**	**67.90**	**64.23**	**45.26**	**45.85**	**56.97▼**
	Gupta's[7]	Decoded(dB)	26.64	22.44	28.64	27.24	25.22	×
		Computations(%)	100	100	100	100	100	×
8×8	Previous[34]	Predicted(dB)	26.50	22.19	28.46	27.17	24.99	×
		Deviation(dB)	**0.14**	**0.25**	**0.18**	**0.07**	**0.23**	**0.17**
		Computations(%)	**67.19**	**74.22**	**67.97**	**56.15**	**50.00**	**63.11**
	Proposed	Predicted(dB)	26.56	22.26	28.52	27.19	25.11	×
		Deviation(dB)	**0.08**	**0.18**	**0.12**	**0.05**	**0.11**	**0.11▼**
		Computations(%)	**55.47**	**66.89**	**59.28**	**49.12**	**43.55**	**54.86▼**

5. Discussion

While the threshold of $\mathrm{RE_{ACEC}}$ is set at 0.15, by Eq.(18), we have:

$$\mathrm{RE_{CACE}}=0.15 \tag{27}$$

Then, we have:

$$\mathrm{ACE^{Coded}}=6.13\times\mathrm{ACE_{Max}^{Uncoded}} \tag{28}$$

While the actual percentage of accumulated collage error (APACE) is defined as

$$\mathrm{APACE}=\frac{\mathrm{ACE^{Coded}}}{\mathrm{ACE^{Coded}}+\mathrm{ACE^{Uncoded}}} \tag{29}$$

Then, the minimum of APACE, $\mathrm{APACE_{Min}}$, can be defined and calculated by Eqs.(29) and (28), respectively, as:

$$\mathrm{APACE_{Min}}=\frac{\mathrm{ACE^{Coded}}}{\mathrm{ACE^{Coded}}+\mathrm{ACE_{Max}^{Uncoded}}}\approx86\% \tag{30}$$

In the previous work, the encoding process terminated while $\mathrm{APACE_{Min}}$ reaches 90%. Thus, we know that $\mathrm{RE_{CACE}}$ set at 0.15 needs fewer computations than $\mathrm{APACE_{Min}}$ set at

90%. In other words, while $\mathrm{RE_{CACE}}$ reaches 0.15, $\mathrm{APACE_{Min}}$ is just about 86% and smaller than 90%, which implies that in the previous method, we need to encode extra 4% range blocks to make $\mathrm{APACE_{Min}}$ reach 90%. Thus, the corresponding computations in the proposed method can be are saved. For example, while the size of range blocks is set at 4 × 4, the green and red blocks in Fig.3 represent the range blocks encoded in the previous method, and the green blocks represent the range blocks encoded in the proposed method. For Bird image, we totally have 4,096 range blocks. In the previous method, while $\mathrm{APACE_{Min}}$ reaches 90%, we need to encode 2,826 range blocks which are represented by the red and green blocks in Fig.3(a) and comprise 68.99% of total range blocks in Tab.1. In the proposed method, while $\mathrm{RE_{CACE}}$ reaches 0.15, we only need to encode 2,489 range blocks which are represented by the green blocks in Fig.3(a) and comprise 60.77% of total range blocks in Tab.1. This implies that the computations of 337 range blocks which are represented by red blocks can be saved. While the size of range blocks is set at 8 × 8, similar results as in Fig.3 were illustrated in Fig.4. For Bird image, we totally have 1,024 range blocks. In the previous method, while $\mathrm{APACE_{Min}}$ reaches 90%, we need to encode 689 range blocks which are represented by the red and green blocks in Fig.4(a) and comprise 67.29% of total range blocks in Tab.1. In the proposed method, when $\mathrm{RE_{CACE}}$ reaches 0.15, we only need to encode 568 range blocks which are represented by the green blocks in Fig.4(a) and comprise 55.47% of total range blocks in Tab.1. This implies that the computations of 121 range blocks which are represented by red blocks can be saved. Moreover, ACEC in the proposed method tries to reflect the proportional relationship among $\mathrm{ACE^{All}}$, $\mathrm{ACE_{Max}^{All}}$ and $\mathrm{ACE_{Min}^{All}}$ which has not been considered in the previous method. Thus, the proposed method can provide

more accurate prediction results. In summary, the proposed method can predict the decoded image quality more accurately with fewer computations than the previous method.
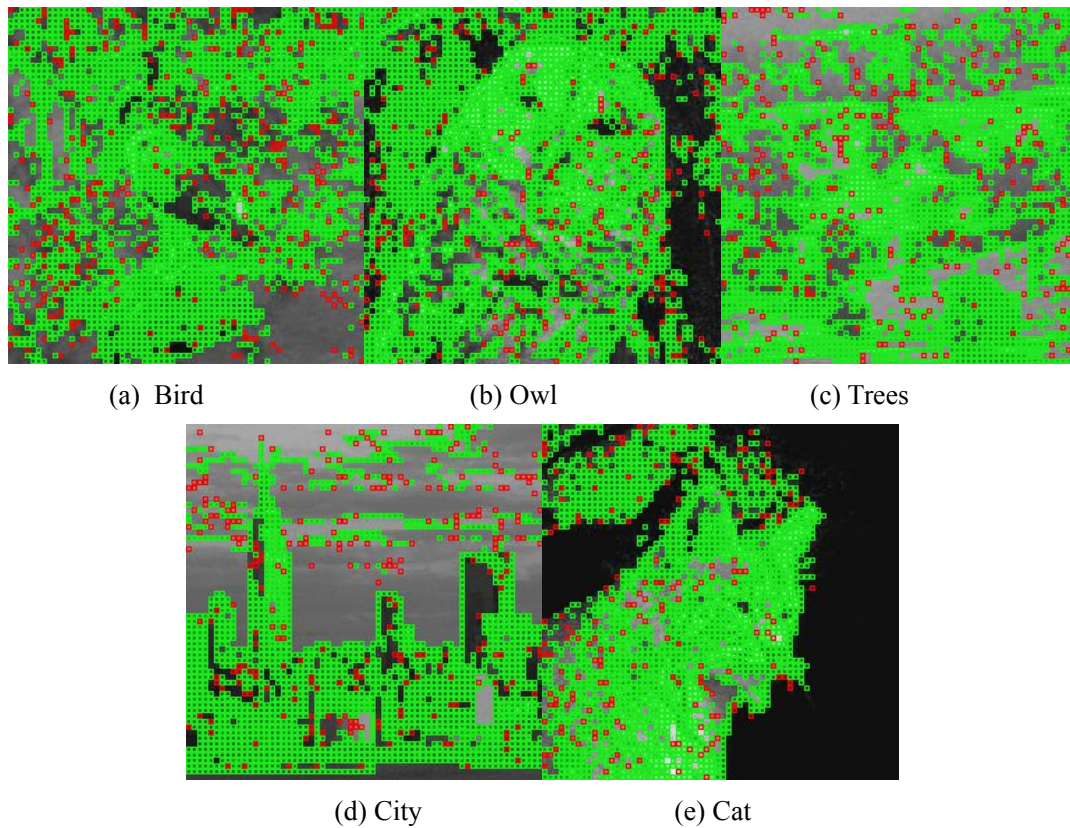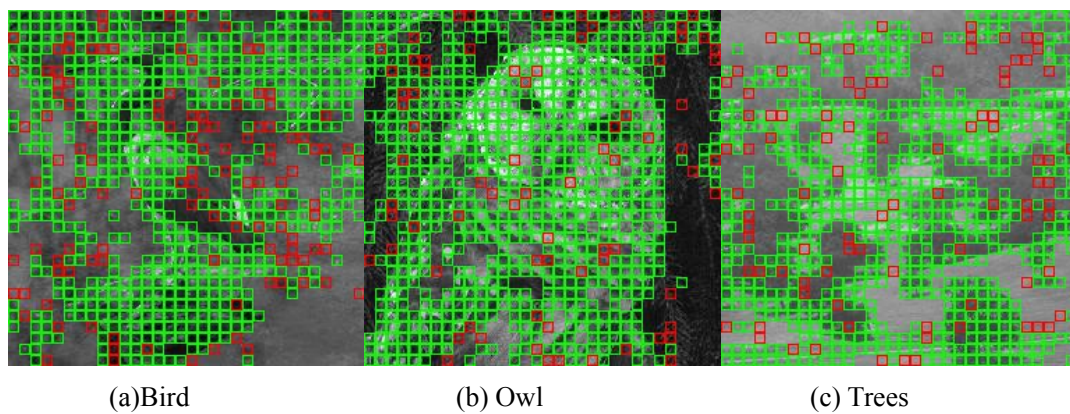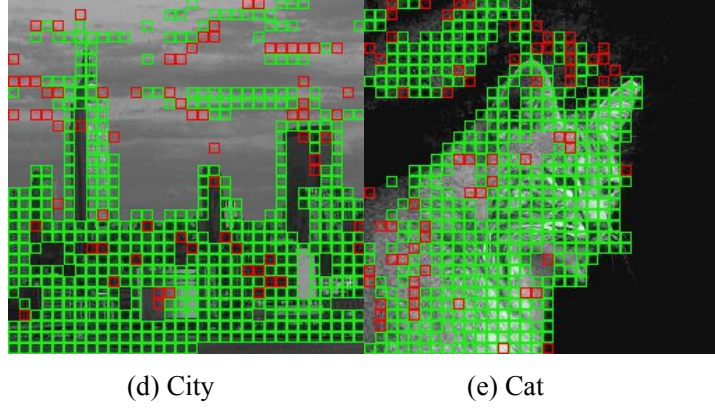

(a) Bird   (b) Owl   (c) Trees

(d) City   (e) Cat

Fig.3 While the size of range blocks is 4×4, the green and red blocks represent the range blocks encoded in the previous method, and the green blocks represent the range blocks encoded in the proposed method.

(a)Bird   (b) Owl   (c) Trees

| (d) City | (e) Cat |

Fig.4 While the size of range blocks is 8×8, the green and red blocks represent the range blocks encoded in the previous method, and the green blocks represent the range blocks encoded in the proposed method.

6. Conclusion

An accumulated collage error coefficient based fractal prediction method was proposed in this study. Firstly, the definition of ACEC was introduced to describe the proportional relationship among the upper bound, lower bound, and actual value of the accumulated collage error. Then, the definition and monotonicity of the relative error of ACEC were also defined and discussed, respectively. Finally, while the relative error of ACEC reaches a small value, the accumulated collage error of all range blocks can be estimated to directly predict the decoded image quality. Experimental results verify that the proposed method can provide higher prediction accuracy with fewer computations than the previous method. In future work, we will attempt to improve the performance of the fractal prediction method in terms of prediction accuracy and the number of computations further.

References

[1] Jacquin AE. Image coding based on a fractal theory of iterated contractive image transformations. IEEE Trans Image Process 1992; 1(1): 18-30.

[2] Fisher Y. Fractal Image Compression: Theory and Application. Springer-Verlag; 1994.

[3] Wohlberg B, Jager G de. A review of the fractal image coding literature. IEEE Trans Image Process 1999; 8(12): 1716-1729.

[4] Jacquin AE. Fractal image coding: A review. Proc IEEE 1993; 81(10): 1451-1465.

[5] Chaurasia V, Chaurasia V. Statistical feature extraction based technique for fast fractal image compression. J Vis Commun Image R 2016; 41: 87-95.

[6] Zheng YP, Li XP, Sarem M. Fast fractal image compression algorithm using specific update search. IET Image Process 2020; 14(9): 1733-1739.

[7] Gupta R, Mehrotra D, Tyagi RK. Hybrid edge-based fractal image encoding using K-NN search. Multimed Tools Appl 2022; 81(15): 21135-21154.

[8] Shen FR, Osamu H. A fast no search fractal image coding method. Signal Process Image Commun 2004; 19(5): 393-404.

[9] Wang XY, Wang SG. An improved no-search fractal image coding method based on a modified gray-level transform. Comput Graph 2008; 32(4): 445-450.

[10] Wang XY, Wang YX, Yun JJ. An improved no-search fractal image coding method based on a fitting plane. Image Vision Comput 2010; 28, 1303-1308.

[11] Bi S, Wang Q. Fractal image coding based on a fitting surface. J Appl Math 2014; 634848.

[12] Ghazel M, Freeman GH, Vrscay ER. Fractal image denoising. IEEE Trans Image Process 2003; 12(12): 1560-1578.

[13] Ghazel M, Freeman GH, Vrscay ER. Fractal-wavelet image denoising revisited. IEEE Trans Image Process 2006; 15(9): 2669-2675.

[14] Lu J, Ye ZX, Zou YR, Ye RS. An enhanced fractal image denoising algorithm. Chaos Solitons & Fractals 2008; 38(4): 1054-1064.

[15] Jeng JH, Tseng CC, Hsieh JG. Study on huber fractal image compression. IEEE Trans Image Process 2009; 18(5): 995-1003.

[16] Lu J, Ye ZX, Zou YY. Huber fractal image coding based on a fitting plane. IEEE Trans Image Process 2013; 22(1): 134-145.

[17] Zou YY, Hu HX, Lu J, Liu XX, Jiang QT, Song GH. A nonlocal low-rank regularization method for fractal image coding. Fractals 2021; 29(5): 2150125.

[18] Xu C, Ye YT, Hu ZW, Zou YR, Shen LX, Liu XX, Liu J. A primal-dual algorithm for robust fractal image coding. Fractals 2019; 27(7): 1950119.

[19] Pi MH, Mandal MK, Basu A. Image retrieval based on histogram of fractal parameters. IEEE Trans

Multimedia 2005; 7(4): 597-605.

[20] Wang XY, Chen Z. A fast fractal coding in application of image retrieval. Fractals 2009; 17(4): 441-450.

[21] Huang X, Zhang Q, Liu W. A new method for image retrieval based on analyzing fractal coding characters. J Vis Commun Image R 2013; 24(1): 42-47.

[22] Chung KH, Fung YH, Chan YH. Image enlargement using fractal. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 2003; VI-273: 273-275.

[23] Lai CM, Lam KM, Chan YH, Siu WC. An efficient fractal based algorithm for image magnification. In: Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, 2004; ii-iii: 571-574.

[24] Chen ZP, Ye ZL, Wang SX, Peng GH. Image magnification based on similarity analogy. Chaos Solitons & Fractals 2009; 40(5): 2370-2375.

[25] Wee YC, Shin HJ. A novel fast fractal super resolution technique. IEEE Trans Consumer Electron 2010; 56(3): 1537-1541.

[26] Hua Z, Zhang HC, Li JJ. Image super resolution using fractal coding and residual network. Complexity 2019; 9419107.

[27] Pi MH, Li CH, Li H. A novel fractal image watermarking. IEEE Trans Multimedia 2006; 8(3): 488-499.

[28] Daraee F, Mozaffari S. Watermarking in binary document images using fractal codes. Pattern Recognit Lett 2014; 35: 120-129.

[29] Lu J, Zou YR, Yang CY, Wang LJ. A robust fractal color image watermarking algorithm. Math Probl Eng 2014: 638174.

[30] Abdullahi SM, Wang HX, Li T. Fractal coding-based robust and alignment-free fingerprint image hashing. IEEE Trans Inform Forensic Secur 2020; 15: 2587-2601.

[31] Khelaifi F, He HJ. Perceptual image hashing based on structural fractal features of image coding and ring partition. Multimed Tools Appl 2020; 79(27-28): 19025-19044.

[32] Bisogni C, Nappi M, Pero C, Ricciardi S. FASHE: A fractal based strategy for head pose estimation. IEEE Trans on Image Process 2021; 30: 3192-3203.

[33] Bisogni C, Nappi M, Pero C, Ricciardi S. PIFS scheme for head pose estimation aimed at faster face

recognition. IEEE Trans on Biometrics, Behavior, and Identity Science 2022; 4: 173-184.

[34] Wang Q, Bi S. Improved method for predicting the peak signal-to-noise ratio quality of decoded images in fractal image coding. J Electron Imaging 2017; 26: 013024.